

Multi-visit Clustered Team Orienteering Problem

S. Hanafi¹, R. Mansini², R. Zanotti²

1 LAMIH, CNRS UMR 8201, University of Valenciennes, France

Said.Hanafi@univ-valenciennes.fr

2 Department of Information Engineering, University of Brescia

renata.mansini@unibs.it, roberto.zanotti@unibs.it

Odysseus 2018, Cagliari 3-8, June 2018

Outline

The Problem

Problem Definition
Literature
Formulation

Model Enforcement

Variables Fixing
Variables Bound
Valid Inequalities

Solution Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

Computational Results

Benchmark instances
Setting
Results
Conclusions

- The Problem
 - Definition
 - Literature Review
 - Mathematical Formulation
- Model Enforcement
 - Variables fixing
 - Variables bound
 - Valid inequalities
- Solution Methods
 - Iterative Kernel Search
 - Primal Kernel Search
 - Branch-and-Cut algorithm
- Computational results
- Conclusions

Multi-visit Clustered Team Orienteering Problem

Outline

The Problem

Problem Definition

Literature

Formulation

Model

Enforcement

Variables Fixing

Variables Bound

Valid Inequalities

Solution

Algorithms

Basic KS

Iterative KS

Primal KS

Branch & Cut

Computational

Results

Benchmark instances

Setting

Results

Conclusions

Given:

- a set of nodes $V = N \cup \{0, n + 1\}$, where $N = \{1, \dots, n\}$ (set of tasks) partitioned into m clusters (customers)
$$N = \bigcup_{h=1}^m C_h$$
- a set of arcs $A = \{(i, j) : i \neq n + 1; j \neq 0; i \neq j; i, j \in V\}$
- a set K of dedicated vehicles (technicians), each one specialized in the execution of a subset of nodes (tasks)
- a time bound T_{max} for each vehicle
- a profit p_h for each cluster $h = 1, \dots, m$
- a service time d_i for each $i \in N$
- a traveling time $t_{ij}, (i, j) \in A$

Outline

The Problem

Problem Definition

Literature

Formulation

Model

Enforcement

Variables Fixing

Variables Bound

Valid Inequalities

Solution

Algorithms

Basic KS

Iterative KS

Primal KS

Branch & Cut

Computational

Results

Benchmark instances

Setting

Results

Conclusions

Constraints:

- each route starts from the depot at time 0
- the traveling time plus the service time of each route cannot exceed T_{max}
- if a cluster is selected all its nodes have to be executed according to a predefined order
- each node of a selected cluster has to be visited exactly once by one vehicle enabled to serve it
- no waiting time allowed to a node

Objective: Find a set of routes, each one not exceeding T_{max} , that maximizes total collected profit while satisfying precedence constraints

Multi-visit Clusters

Outline

The Problem

Problem Definition

Literature

Formulation

Model

Enforcement

Variables Fixing

Variables Bound

Valid Inequalities

Solution

Algorithms

Basic KS

Iterative KS

Primal KS

Branch & Cut

Computational

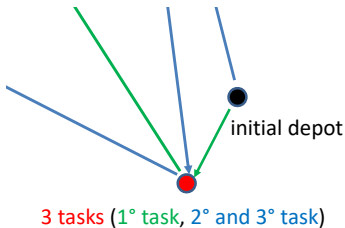
Results

Benchmark instances

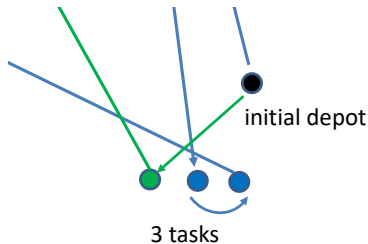
Setting

Results

Conclusions



Nodes as clusters



Nodes as tasks

Real Case Application

Outline

The Problem

Problem Definition

Literature

Formulation

Model

Enforcement

Variables Fixing

Variables Bound

Valid Inequalities

Solution

Algorithms

Basic KS

Iterative KS

Primal KS

Branch & Cut

Computational

Results

Benchmark instances

Setting

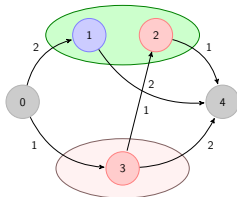
Results

Conclusions

Large company assembling equipments and furniture on site. Tasks are accomplished by technicians working in team.

Request: Evaluate the advantage to let technicians move independently among customers

Assumption: No waiting times at nodes



2 vehicles, 2 clusters, 3 nodes (service time $d_1 = 2; d_2 = 1; d_3 = 1$), $T_{max} = 6$

No optimal solution by forcing the use of two vehicles.

Outline

The Problem

Problem Definition

Literature

Formulation

Model

Enforcement

Variables Fixing

Variables Bound

Valid Inequalities

Solution

Algorithms

Basic KS

Iterative KS

Primal KS

Branch & Cut

Computational

Results

Benchmark instances

Setting

Results

Conclusions

■ Clustered routing problems:

– Clustered TSP:

Chisman [1975]

– Clustered VRP:

Sevaux and Sörensen [2008]

■ Clustered Orienteering Problems:

– Single vehicle:

Angelelli et al. [2014] (Branch-and-Cut, Tabu Search)

Yahiaoui et al. [2017] (Hybrid heuristic)

– Multi-vehicle:

no contributions

Problem Formulation 1/2

Outline

The Problem

Problem Definition

Literature

Formulation

Model

Enforcement

Variables Fixing

Variables Bound

Valid Inequalities

Solution

Algorithms

Basic KS

Iterative KS

Primal KS

Branch & Cut

Computational

Results

Benchmark instances

Setting

Results

Conclusions

$$\max \sum_{h=1}^m p_h y_h \quad (1)$$

$$\sum_{(0,i) \in \delta^+(0)} x_{0i}^k = \sum_{(i,n+1) \in \delta^-(n+1)} x_{i,n+1}^k = 1 \quad k \in K \quad (2)$$

$$\sum_{(i,j) \in \delta^+(i)} x_{ij}^k = \sum_{(j,i) \in \delta^-(i)} x_{ji}^k = u_i^k \quad i \in N, k \in K_i \quad (3)$$

$$\sum_{k \in K_i} u_i^k = y_h \quad i \in C_h, h = 1, \dots, m, \quad (4)$$

$$x_{ij}^k \in \{0, 1\} \quad (i, j) \in A, k \in K \quad (5)$$

$$u_i^k \in \{0, 1\} \quad i \in N, k \in K_i \quad (6)$$

$$y_h \in \{0, 1\} \quad h = 1, \dots, m \quad (7)$$

Problem Formulation 2/2

Outline

The Problem

Problem Definition

Literature

Formulation

Model

Enforcement

Variables Fixing

Variables Bound

Valid Inequalities

Solution

Algorithms

Basic KS

Iterative KS

Primal KS

Branch & Cut

Computational

Results

Benchmark instances

Setting

Results

Conclusions

$$\sum_{(i,j) \in \delta^+(i)} z_{ij} - \sum_{(j,i) \in \delta^-(i)} z_{ji} = \sum_{k \in K} \sum_{(i,j) \in \delta^+(i)} (t_{ij} + d_i) x_{ij}^k \quad i \in N \quad (8)$$

$$z_{0i} = t_{0i} \sum_{k \in K} x_{0i}^k \quad i \in N \quad (9)$$

$$z_{ij} \geq (t_{0i} + t_{ij} + d_i) \sum_{k \in K} x_{ij}^k \quad (i,j) \in A \setminus \{(0, n+1)\} \quad (10)$$

$$z_{ij} \leq (T_{max} - d_j - t_{j,n+1}) \sum_{k \in K} x_{ij}^k \quad (i,j) \in A \setminus \{(0, n+1)\} \quad (11)$$

$$\sum_{(s,i) \in \delta^-(i)} z_{si} + d_i \sum_{k \in K} u_i^k \leq \sum_{(s,j) \in \delta^-(j)} z_{sj} \quad i \in \bar{C}_h, j = i+1, h = 1, \dots, m \quad (12)$$

$$z_{ij} \geq 0 \quad (i,j) \in A \setminus \{(0, n+1)\} \quad (13)$$

Maffioli and Sciomachen: A mixed-integer model for solving ordering problems with side constraints [1997]

Model Enforcement: Variables Fixing

Outline

The Problem

Problem Definition
Literature
Formulation

Model Enforcement

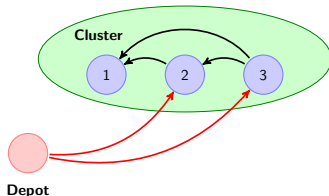
Variables Fixing
Variables Bound
Valid Inequalities

Solution Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

Computational Results

Benchmark instances
Setting
Results
Conclusions



- $x_{0j}^k = 0 \quad j \in N_k : \Gamma_j^- \neq \emptyset$
- $x_{ij}^k = 0 \quad i, j \in N_k, j \in \Gamma_i^-, k \in K_i \cap K_j$

$\Gamma_j^- = \{\text{set of nodes preceding } j \text{ in the same cluster}\}$
 $N_k = \{\text{set of nodes that can be served by vehicle } k\}$
 $K_i = \{\text{set of vehicles that can serve } i\}$

Model Enforcement: Variables Bound

Outline

The Problem

Problem Definition
Literature
Formulation

Model Enforcement

Variables Fixing
Variables Bound
Valid Inequalities

Solution Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

Computational Results

Benchmark instances
Setting
Results
Conclusions

Lower bounds enforcement:

For each $(i, j) \in A \setminus \{(0, n+1)\}$:

$$z_{ij} \geq \max\{t_{0i} + t_{ij} + d_i + \sum_{u \in \Gamma_i^-} d_u; t_{0j} + \sum_{u \in \Gamma_j^-} d_u\} \sum_{k \in K} x_{ij}^k$$

Upper bounds enforcement:

For each $(i, j) \in A \setminus \{(0, n+1)\}$:

$$z_{ij} \leq (T_{max} - d_j - t_{j,n+1} - \sum_{u \in \Gamma_j^+} d_u) \sum_{k \in K} x_{ij}^k$$

Valid Inequalities: Global Time Cuts

Outline

The Problem

Problem Definition
Literature
Formulation

Model

Enforcement

Variables Fixing
Variables Bound
Valid Inequalities

Solution

Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

Computational

Results

Benchmark instances
Setting
Results
Conclusions

For each vehicle $k \in K$, the following inequality holds:

$$\sum_{(i,j) \in A} (t_{ij} + d_j)x_{ij}^k \leq T_{max}$$

Separation algorithm:

We add these inequalities directly to the formulation
(no separation)

Valid Inequalities: Connectivity Cuts (CC)

Outline

The Problem

Problem Definition

Literature

Formulation

Model

Enforcement

Variables Fixing

Variables Bound

Valid Inequalities

Solution

Algorithms

Basic KS

Iterative KS

Primal KS

Branch & Cut

Computational

Results

Benchmark instances

Setting

Results

Conclusions

$$\sum_{(i,j) \in \delta^+(S)} x_{ij}^k \geq u_r^k \quad S \subseteq N, r \in S, k \in K$$

Separation algorithm:

- Exact solution of a minimum cut problem by using GUROBI
- Problem formulated for each vehicle $k \in K$ on a graph $\bar{G}_k = (V, \bar{A}_k)$ induced by the optimal solution of LP relaxation
- LP value of variables x_{ij}^k used as capacity of the arcs

Valid Inequalities: Sequentiality Cuts (SC)

Outline

The Problem

Problem Definition
Literature
Formulation

Model

Enforcement

Variables Fixing
Variables Bound

Valid Inequalities

Solution

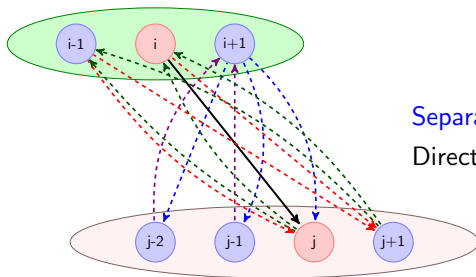
Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

Computational

Results

Benchmark instances
Setting
Results
Conclusions



Separation algorithm:
Direct check

$$\begin{aligned}
 & \sum_{t \in C_h, t < i} \sum_{u \in C_l, u \geq j} x_{tu}^k + \sum_{u \in C_l, u > j} x_{iu}^k + \sum_{t \in C_h, t > i} \sum_{u \in C_l, u \leq j} x_{tu}^k + \\
 & + \sum_{t \in C_l, t \geq j} \sum_{u \in C_h, u \leq i} x_{tu}^k + \sum_{t \in C_l, t < j} \sum_{u \in C_h, u > i} x_{tu}^k \leq 2 \min\{|C_h|; |C_l|\} (1 - x_{ij}^k)
 \end{aligned}$$

Kernel Search 1/2

Outline

The Problem

Problem Definition
Literature
Formulation

Model

Enforcement

Variables Fixing
Variables Bound
Valid Inequalities

Solution

Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

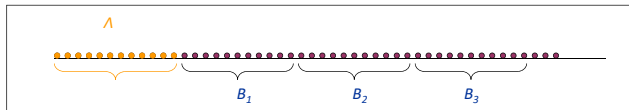
Computational

Results

Benchmark instances
Setting
Results
Conclusions

Main features (Angelelli, Mansini, Speranza [2010], [2012]):

- 1** **Sorting** of items (variables) according to a predefined criterion (variable LP values, reduced costs)
- 2** **Identification** of a **Kernel Set** (Λ) of promising variables (more probably selected in an optimal solution) and **partition** of remaining variables into buckets



- 3** **Construction and solution** of a sequence of **small subproblems** by a MIP solver:

$$\text{Subproblem} = \text{MIP}(\Lambda \cup B) \quad B \in \mathcal{B}$$

Outline

The Problem

Problem Definition
Literature
Formulation

Model

Enforcement

Variables Fixing
Variables Bound
Valid Inequalities

Solution

Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

Computational Results

Benchmark instances
Setting
Results
Conclusions

Critical aspects:

- Manage trade-off between solution quality and computational effort (appropriate size of Kernel Set and buckets)
- Find alternatives to reduced costs when they do not provide a clear signal on promising variables
- Find an effective way for variables to enter/leave the Kernel Set after its initial construction
- Optimize the breakdown of the global solution time used by the method

Adaptive Kernel Search

(Guastaroba, Savelsbergh, Speranza [2017])

New Iterative Kernel Search

Outline

The Problem

Problem Definition
Literature
Formulation

Model

Enforcement

Variables Fixing
Variables Bound
Valid Inequalities

Solution

Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

Computational Results

Benchmark instances
Setting
Results
Conclusions

Iterative variant: scroll the buckets more than once (bucket iterations)

Innovative aspects:

- 1** Two sorting rules that can be interchanged in the bucket iterations
- 2** Dynamic procedure to eliminate variables from Kernel Set (using information from solved subproblems)
- 3** Redefinition of Kernel Set and bucket size at the beginning of each bucket iteration
- 4** Dynamic adjustment of subproblems solution time

Variables Sorting: First Iteration

Outline

The Problem

Problem Definition
Literature
Formulation

Model

Enforcement

Variables Fixing
Variables Bound
Valid Inequalities

Solution

Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

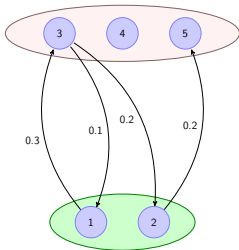
Computational Results

Benchmark instances
Setting
Results
Conclusions

- 1 Compute for each cluster h (max number of iterations $iter$):

$$score[h] \leftarrow score[h] + \chi_1 \frac{p_h}{maxProfit} + \chi_2 \frac{maxSize - |C_h|}{maxSize}$$

- 2 Sort clusters in non-increasing order of their score
- 3 Compute *affinity level* for each pair of clusters and adjust sorting:



Variables Sorting: Successive Iterations

Outline

The Problem

- Problem Definition
- Literature
- Formulation

Model

Enforcement

- Variables Fixing
- Variables Bound
- Valid Inequalities

Solution

Algorithms

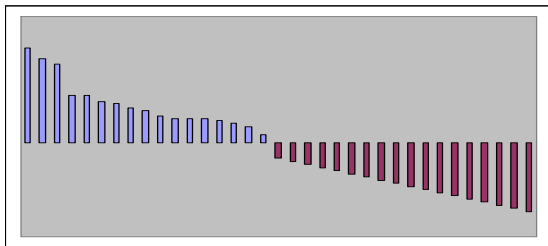
- Basic KS
- Iterative KS
- Primal KS
- Branch & Cut

Computational

Results

- Benchmark instances
- Setting
- Results
- Conclusions

Sorting depending on **variables and reduced costs value** in the optimal solution of the LP relaxation:



Buckets and Kernel Set Construction

Outline

The Problem

Problem Definition
Literature
Formulation

Model

Enforcement

Variables Fixing
Variables Bound
Valid Inequalities

Solution

Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

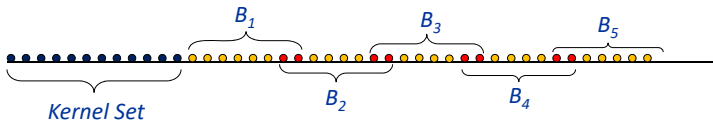
Computational

Results

Benchmark instances
Setting
Results
Conclusions

At the beginning of each bucket iteration:

- **Kernel Set** encompasses all clusters of the **best integer solution found so far** plus the first ξ clusters in the new sorting
- Remaining clusters are divided into buckets of **equal size** (but reduced with respect to previous iteration) that **partially overlap**:



Varying Kernel Set

Outline

The Problem

Problem Definition
Literature
Formulation

Model

Enforcement

Variables Fixing
Variables Bound
Valid Inequalities

Solution

Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

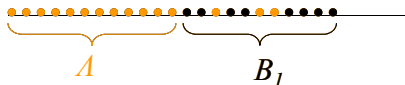
Computational

Results

Benchmark instances
Setting
Results
Conclusions

– Insert procedure:

- SOLVESUBPROBLEM($\Lambda \cup B$) for each $B \in \mathcal{B}$
- Add to Λ all clusters selected in the subproblem solution



– Eject procedure:

- If $e_h - s_h \geq \text{threshold}$, remove cluster C_h from Λ

$\mathcal{S}_h = \{\text{set of solutions found by KS since cluster } C_h \text{ entry in } \Lambda\}$

s_h (e_h) number of times C_h belongs (does not) to solutions in \mathcal{S}_h

Outline

The Problem

Problem Definition
Literature
Formulation

Model

Enforcement

Variables Fixing
Variables Bound
Valid Inequalities

Solution

Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

Computational Results

Benchmark instances
Setting
Results
Conclusions

For each iteration:

- Kernel Set: $(1+\beta)$ (Global time iteration)/ $(\#buckets+1)$
- Each Bucket: $(1+\beta)$ (Residual time)/ $(residual \#buckets)$

Motivation: Kernel Set and initial buckets should contain more promising clusters and thus corresponding subproblems should receive higher computational time

Primal Kernel Search (PKS)

Outline

The Problem

Problem Definition
Literature
Formulation

Model

Enforcement

Variables Fixing
Variables Bound
Valid Inequalities

Solution

Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

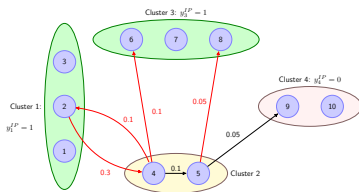
Computational

Results

Benchmark instances
Setting
Results
Conclusions

Only Kernel Set + 1 Bucket

- **Sorting:** exploit information provided by local LP relaxation:
 - Affinity on groups of clusters



- **Memorization procedure:**
 - load active clusters and solution for each subproblem solved
 - if a subproblem has already been solved in a previous call, skip it and solve the next one

Outline

The Problem

Problem Definition
Literature
Formulation

Model

Enforcement

Variables Fixing
Variables Bound
Valid Inequalities

Solution

Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

Computational Results

Benchmark instances
Setting
Results
Conclusions

Dynamic separation of cuts: implemented rules

- **Connectivity Cuts:** checked at each node of the search tree (root included). Added for $s^* = \operatorname{argmax}_{s \in S} \{\bar{u}_s^k\}$, only if its violation is greater than ϵ_1 .
- **Sequentiality Cuts:** checked at each node of the search tree (root included). Added only the first 3 violated inequalities with violation greater than ϵ_2 .

Benchmark Problems

Outline

The Problem

Problem Definition
Literature
Formulation

Model

Enforcement

Variables Fixing
Variables Bound
Valid Inequalities

Solution

Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

Computational

Results

Benchmark instances
Setting
Results
Conclusions

Graphs and profits taken from Chao et al. [1997]:
Sets 4,5,6,7.

For each graph:

- 3 random generations of # tasks and service times for each cluster
- 3 numbers of vehicle ($|K| = 2,3,4$)
- 3 values of T_{max}

Per Class: 27 instances

Total: 108 instances (plus a few real case instances)

Experimental Setting

Outline

The Problem

Problem Definition
Literature
Formulation

Model

Enforcement

Variables Fixing
Variables Bound
Valid Inequalities

Solution

Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

Computational

Results

Benchmark instances

Setting

Results

Conclusions

- **HW:** 64-bit Windows Personal Computer, 2.93 GHz Intel i7 870 processor, and 8 GB of RAM.
- **SW:** Java 8, GUROBI 7.5.2: all built-in cuts activated, 8 threads, Barrier Algorithm

Exact Algorithms: 2 hours time limit

- BASIC: Compact formulation solved with GUROBI
- B&C: Branch-and-Cut with dynamic separation of connectivity and sequentiality cuts, bounds enforcement, variables fixing
- B&C-PKS: B&C plus Primal Kernel Search

Heuristic Algorithms:

- Iterative KS 15 min.: 10 (first iteration) + 5 (second iteration)
- Primal KS 1 min.: 30 secs (Kernel Set), 30 secs (first bucket)

Instance	BASIC			B&C					B&C-PKS						
	Obj	Ttb	gap	Obj	Ttb	CC	SC	gap	Obj	Ttb	CC	SC	#PKS	#Imp	gap
p5.2.s0	40	33	1214.0	340	6999	1760	1764	47.3	465	2575	395	152	53	5	4.12
p5.2.s0t2	125	7005	366.0	445	4290	1331	804	26.8	515	2635	492	225	40	7	4.81
p5.2.s0t3	250	7203	153.0	280	3599	2169	1924	119.0	465	4963	1132	931	35	8	31.1
p5.2.s1	255	7203	165.0	445	6791	2054	1425	17.7	465	4747	1179	527	21	3	10.6
p5.2.s1t2	245	457	205.0	330	3788	1088	571	76.4	510	2742	1195	1251	27	7	13.1
p5.2.s1t3	255	4639	218.0	395	7104	646	220	60.4	585	5164	1423	713	31	6	6.88
p5.2.s2	90	5460	510.0	350	4358	1119	877	49.2	420	6996	1226	1022	44	9	24.8
p5.2.s2t2	320	6154	91.2	305	3098	1885	1509	88.9	385	5379	1233	613	27	4	48.9
p5.2.s2t3	170	7036	284.0	100	2362	1973	2207	535.0	465	6779	1131	1118	32	7	34.0
p5.3.s0	60	1530	782.0	250	1264	407	45	72.1	305	6955	884	237	25	5	42.4
p5.3.s0t2	55	4158	974.0	285	3320	516	34	71.5	340	5230	1855	637	33	6	41.4
p5.3.s0t3	65	5292	895.0	240	6493	2538	954	128.0	315	6911	930	297	22	5	70.2
p5.3.s1	220	7173	194.0	275	7202	444	21	50.7	215	3019	237	19	14	6	89.7
p5.3.s1t2	235	472	206.0	140	6243	592	34	232.0	245	6766	293	29	8	2	85.9
p5.3.s1t3	255	5083	208.0	140	4234	1007	66	266.0	315	2749	363	37	10	4	61.9
p5.3.s2	80	53	570.0	305	6797	1710	495	46.9	350	4677	329	67	29	8	24.9
p5.3.s2t2	195	487	207.0	280	4342	605	36	75.8	350	7204	763	96	22	6	39.7
p5.3.s2t3	200	1784	231.0	280	4584	735	48	93.5	355	5311	1581	734	27	6	52.4
p5.4.s0	145	7198	262.0	305	2643	369	54	55.3	360	4886	275	34	40	7	31.2
p5.4.s0t2	145	3927	311.0	100	3825	418	36	439.0	355	4326	1249	250	34	7	52.5
p5.4.s0t3	180	5319	273.0	125	4133	1154	50	382.0	360	6408	164	10	21	8	67.5
p5.4.s1	185	6808	213.0	245	2782	283	14	83.2	330	7406	238	18	26	6	36.1
p5.4.s1t2	185	266	255.0	170	3888	412	14	203.0	275	3539	950	25	21	3	85.5
p5.4.s1t3	200	1972	266.0	240	7197	516	12	138.0	280	1772	682	31	18	4	104.0
p5.4.s2	140	6357	245.0	255	3698	321	21	64.1	330	1916	496	42	25	4	24.7
p5.4.s2t2	85	857	544.0	275	4737	1334	122	69.5	255	3597	2068	562	20	2	79.2
p5.4.s2t3	135	1132	355.0	230	5703	801	17	122.0	275	5717	960	54	21	4	85.5
	167.2	3891.0	377.7	264.1	4647.2	1044.0	495.3	133.8	366.1	4828.5	878.6	360.4	26.9	5.5	46.4

Instance	IKS										B&C+PKS		
	Obj	Ttb	#Sub	avKS	maxKS	avSub	maxSub	SOL	Implt1	Implt2-3	Obj	Ttb	Δ%
p5.2.s0	455	807	22	17.3	24	23.8	30	17	7	4	465	2575	2.2
p5.2.s0t2	485	620	16	15.8	21	22.8	29	17	7	3	515	2635	5.8
p5.2.s0t3	495	577	16	15.4	19	22.4	27	19	7	1	465	4963	-6.5
p5.2.s1	480	485	21	18.0	21	24.8	29	20	8	1	465	4747	-3.2
p5.2.s1t2	465	527	16	15.7	19	22.6	27	19	7	0	510	2742	8.8
p5.2.s1t3	555	468	17	18.6	23	25.4	31	23	9	0	585	5164	5.1
p5.2.s2	460	792	21	16.1	21	23.2	29	18	7	5	420	6996	-9.5
p5.2.s2t2	510	700	21	18.3	22	25.1	30	22	6	4	385	5379	-32.5
p5.2.s2t3	505	454	16	16.2	20	23.1	28	19	8	0	465	6779	-8.6
p5.3.s0	325	372	17	12.8	15	19.5	23	15	7	0	305	6955	-6.6
p5.3.s0t2	350	391	17	11.8	14	18.6	22	14	7	0	340	5230	-2.9
p5.3.s0t3	430	267	16	14.4	18	21.4	26	18	7	0	315	6911	-36.5
p5.3.s1	340	364	16	13.4	17	20.4	25	16	7	0	215	3019	-58.1
p5.3.s1t2	355	548	16	12.6	16	19.8	24	15	9	0	245	6766	-44.9
p5.3.s1t3	380	901	17	10.8	12	17.7	20	16	6	1	315	2749	-20.6
p5.3.s2	345	572	16	13.8	17	20.8	25	17	7	1	350	4677	1.4
p5.3.s2t2	370	562	16	14.8	19	21.7	27	18	7	0	350	7204	-5.7
p5.3.s2t3	395	842	16	13.4	16	20.5	24	17	8	1	355	5311	-11.3
p5.4.s0	340	398	17	12.0	14	18.8	22	14	7	0	360	4886	5.6
p5.4.s0t2	430	625	16	16.1	20	23.1	28	20	8	1	355	4326	-21.1
p5.4.s0t3	405	601	16	15.6	20	22.5	28	19	8	0	360	6408	-12.5
p5.4.s1	330	785	17	13.2	15	19.9	23	16	7	1	330	7406	0.0
p5.4.s1t2	310	489	17	11.5	14	18.4	22	14	6	0	275	3539	-12.7
p5.4.s1t3	345	574	16	13.1	17	20.2	25	17	7	0	280	1772	-23.2
p5.4.s2	330	523	16	14.2	19	21.3	27	18	7	1	330	1916	0.0
p5.4.s2t2	360	420	16	13.7	18	20.7	26	18	8	0	255	3597	-41.2
p5.4.s2t3	355	542	16	13.4	17	20.5	25	17	7	0	275	5717	-29.1
	403.9	563.2	17.0	14.5	18.1	21.4	26.0	17.5	7.3	0.9	366.1	4828.5	-13.3

Conclusions and Future Developments

Outline

The Problem

Problem Definition
Literature
Formulation

Model

Enforcement

Variables Fixing
Variables Bound
Valid Inequalities

Solution

Algorithms

Basic KS
Iterative KS
Primal KS
Branch & Cut

Computational

Results

Benchmark instances
Setting
Results

Conclusions

Already obtained:

- an effective heuristic (IKS) for the solution of the problem
- an effective primal heuristic (PKS)
- a Branch-and-Cut algorithm outperforming plain GUROBI when solving the compact formulation in 2 hours time limit

To be done:

- other valid inequalities
- final refinement for IKS
- validation of PKS as primal heuristic in a MIP solver
- new variant of the problem including waiting times